h

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/842,387	04/25/2001	Daniel D. Grove	SIA-P047	6789

7590 03/12/2004

FERNANDEZ & ASSOCIATES, LLP
PATENT ATTORNEYS
PO BOX D
MENLO PARK, CA 94026-6204

EXAMINER	
TRUONG, CAM Y T	
ART UNIT	**PAPER NUMBER**
2172	5

DATE MAILED: 03/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

4

Office Action Summary	Application No. 09/842,387	Applicant(s) GROVE ET AL.	
	Examiner Cam Y T Truong	Art Unit 2172	

-- The MAILING DATE f this communication appears on the c ver sheet with the c rrespondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 December 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-10 and 12-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-10, 12-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant has amended claims 1-10, 12, 14-27, 29-30, 32-35, 37-40, 42-48 and canceled claim 11 in the amendment filed on 12/9/03. Claims 1-10, 12-48 are pending in this Office Action.

Applicant's arguments with respect to claim 46 have been considered but are moot in view of the new ground(s) of rejection.

Applicant argued that Mark Allen Weiss does not teach the claimed limitation "a head representing a first pointer to a first leaf node of a tree structure separate from the data structure; a tail representing a second pointer to a second leaf node of the tree structure; and a body, physically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes of the tree structure" as recited in claims 1 and 6.

However, Mark Allen Weiss teaches the claimed limitations:

"a head representing a first pointer to a first leaf node of a tree structure separate from data structure; a tail representing a second pointer to a second leaf node of the tree structure" as a left binary tree in fig. 17.10, the rood node has two pointers, the first pointer points to left leaf node. The second pointer points to right leaf node. The first pointer of root node of the binary tree that is presented as a head. This head separates from the right binary tree on fig. 18. 3. The right tree is represented as data structure. The second pointer of root node is presented as a tail (page 516, lines 1-22; page 512, fig. 17.4, page 544);

"a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes of the tree structure" as shown in fig. 17.10, a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL). The above information shows that each leaf nodes can be empty node, which stores no data (page 516, lines 1-22; page 517, lines 11-17).

For the above reason, examiner believed that rejection for claims 1-10, 12-45 and 47-48 of the last office action was proper.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-10, 12-45 and 47-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mark ALLen Weiss, "Algorithms, Data Structures, and Problem Solving With C++", (hereinafter "Weiss").

Claims 1, Weiss teaches the claimed limitations:

"a head representing a first pointer to a first leaf node of a tree structure separate from data structure; a tail representing a second pointer to a second leaf node of the tree structure" as a left binary tree in fig. 17.10, the rood node has two pointers, the first

pointer points to left leaf node. The second pointer points to right leaf node. The first pointer of root node of the binary tree that is presented as a head. This head separates from the right tree on fig. 18. 3. The right tree is represented as data structure. The second pointer of root node is presented as a tail (page 516, lines 1-22; page 512, fig. 17.4, page 544).

Weiss does not explicitly teach the claimed limitation "a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes of the tree structure". However, Weiss teaches as shown in fig. 17.10, a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL). The above information shows that each leaf nodes can be empty node, which stores no data (page 516, lines 1-22; page 517, lines 11-17).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Weiss's teaching of a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL) in order to insert new data in tree or sort data tree following user's desire.

As to claim 2, Weiss teaches the claimed limitation "data of the same type" as (page 517, lines 23-60).

As to claim 3, Weiss teaches the claimed limitation "wherein nodes form a sorted tree structure" as (page 642, fig. 20.1).

As to claim 4, Weiss teaches the claimed limitation " wherein the nodes are indexed" as (page 642, fig. 20.1).

As to claim 5, Weiss teaches the claimed limitation " wherein each of the first and the second leaf node comprises a number of data segments" as (page 517, lines 23-60).

As to claim 6, Weiss teaches the claimed limitations:

"providing a sorted tree structure" as shown in fig. 20.3 on, the left tree, which is a heap, is sorted by following the heap order property. This property indicates parent node <= child (page 642);

"preparing a redistribution data structure separate from the sorted tree structure, said redistribution data structure having a head representing a first pointer to a first leaf node of the sorted tree structure" as providing a redistribution data structure such as the right binary tree on the left after deletion of node 5 with one child. This right binary tree is separate from the left binary tree. The right binary tree has a first pointer to a first leaf node of the left binary tree. The left binary tree is presented as the sorted tree structure (page 516, lines 1-22, page 544, fig. 18. 3);

"a tail representing a second pointer to a second leaf node of the sorted tree structure" as (page 516, lines 1-22);

"inserting of a data segment into the tree structure" as (page 647);

"a redistributing of the contiguous empty nodes by employing the redistribution data structure, to enable a more rapid insertion of the data segments" as the each node of the binary tree corresponds to an element of the array that stores the value in the node. Fig. 20-12 shows the routines that add items into the heap. The Toss routine is short; it just adds the new element x in next available location. Insert implements the percolate up using a very tight loop. The for loop that begins at 31 is (x <array [hold/2]; Hole /=2) increments the current size and sets the hole, which is represented as empty node, to the newly added node. The system iterates as long as the item in the parent node is larger than x. Line 32 moves the item in the parent down into the hole, and then the third expression in the for loop moves the hole up to the parent. When the loop terminates, line 33 places x into the hole. The above information shows that the system generates an array for each binary tree to redistribute any node in the binary tree for inserting new item in correct position. Whenever the system wants to insert many items, the system adds many holes as empty nodes into tree by using an array as a data structure to redistribute nodes. An array requires that some operation use linear time. Thus, the array can allow a more rapid insertion of items in a binary tree (page 641; pages 647-648; page 639, lines 19-22).

Weiss does not explicitly teach the claimed limitation "a body, logically adjacent to the head and to the tail, having a set of pointers pointing to contiguous empty nodes

of the sorted tree structure" However, Weiss teaches as shown in fig. 17.10, a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL). The above information shows that each leaf nodes can be empty node, which stores no data (page 516, lines 1-22; page 517, lines 11-17).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Weiss's teaching of a left binary tree has seven nodes, each node has two pointers to point to their leaf nodes. Several efficient implementations of priority queues use trees, BinaryNode (): left (NULL), right (NULL) in order to insert new data in tree or sort data tree following user's desire.

As to claim 7, Weiss teaches the claimed limitation "wherein the data segments is inserted in any order" as (page 551, lines 8-21).

As to claim 8, Weiss teaches the claimed limitation "wherein the sorted tree structure comprises non-leaf and leaf nodes" as (page 551, lines 8-21).

As to claim 9, Weiss teaches the claimed limitation "wherein nodes of the sorted tree structure are indexed" as (page 641, fig. 20.1).

As to claim 10, Weiss teaches the claimed limitation "wherein each of the first and second leaf nodes comprises a number of data segments" as (page 551, lines 8-20).

As to claim 12, Weiss teaches the claimed limitation "the data structure traversing the sorted tree structure in one of a first direction and a second direction" as (page 526, lines 1-10).

As to claim 13, Weiss teaches the claimed limitation "the first direction comprises a-logical, one, and the second direction comprises a logical zero" as (page 526, lines 1-9).

As to claim 14, Weiss teaches the claimed limitation "moving its head one leaf node towards one of the first and second directions" as (page 526, lines 1-9).

As to claim 15, Weiss teaches the claimed limitation "the first direction is towards non-decreasing indices" as (page 526, lines 13-22; page 641, fig. 20.1).

As to claim 16, Weiss teaches the claimed limitation "the second direction is towards non-increasing indices" as (page 526, lines 1-10; page 641, fig. 20.1).

As to claim 17, Weiss teaches the claimed limitation "wherein the redistribution data structure traverses the sorted tree structure when two conditions are met" as (page 648, lines 1-34).

As to claim 18, Weiss teaches the claimed limitation " wherein a first of the two conditions comprises a maximum threshold of filled spaces in the sorted tree structure, and a second of the two conditions comprises a minimum threshold of filled spaces in the sorted tree structure" as (page 644, lines 52-58; page 646, lines 5-8).

As to claim 19, Weiss teaches the claimed limitation "wherein the two conditions are empirically determined" as (page 644, lines 52-58; page 646, lines 5-8).

As to claim 20, Weiss teaches the claimed limitation " wherein the redistribution data structure traverses the sorted tree structure by moving one leaf node towards its traveling direction" as (page 550, lines 1-16).

As to claim 21, Weiss teaches the claimed limitation "an empty leaf node" as (page 550).

As to claim 22, Weiss teaches the claimed limitation "wherein certain conditions are met and the step of redistributing the contiguous empty nodes continues" as (page 551, lines 1-21).

As to claims 23 and 43, Weiss teaches the claimed limitation "wherein the certain conditions are empirically calculated" as (page 551, lines 1-21).

As to claims 24 and 44, Weiss teaches the claimed limitation "wherein the step of redistributing the contiguous empty nodes halts" (page 551, lines 1-21).

As to claims 25, 35 and 45, Weiss teaches the claimed limitation "wherein a data segment insertion restarts the step of redistributing the contiguous empty nodes from where it was last halted" as (page 551, lines 1-21).

As to claim 26, Weiss teaches the claimed limitation "a non-empty leaf node" as if T=Null (page 551, line 10).

As to claim 27, Weiss teaches the claimed limitation "wherein the redistribution data structure copies contents of the head into the tail" as (page 647, lines 1-11).

As to claims 28 and 36, Weiss teaches the claimed limitation "wherein the redistribution data structure travels towards non-decreasing indices" as (page 526, lines 1-10; page 641, fig. 20.1).

As to claims 29 and 39, Weiss teaches the claimed limitation "wherein the sorted tree structure updates from leaf node level to root node level" as (fig. 20. 1, page 646).

As to claims 30 and 40, Weiss teaches the claimed limitation "wherein the contents of the head are cleared and the tail is moved a pre-calculated increment towards non-decreasing indices" as (page 650, lines 1-21).

As to claims 31 and 41, Weiss teaches the claimed limitation "wherein the increment is empirically determined" as (page 651, lines 1-22).

As to claims 32 and 42, Weiss teaches the claimed limitation "wherein certain conditions are met and the step of redistributing the contiguous empty nodes continues" as (page 650, lines 1-21, page 651, lines 1-22).

As to claim 33, Weiss teaches the claimed limitation "wherein the certain conditions are empirically calculated" as (page 650, lines 1-21, page 651, lines 1-22).

As to claim 34, Weiss teaches the claimed limitation " wherein the step of redistributing the contiguous empty nodes halts" as (page 651, lines 1-22).

As to claim 37, Weiss teaches the claimed limitation "wherein the sorted tree structure updates between the tail and the nearest non-empty leaf node whose index is greater than the index of the tail" as fig. 20.14-20.15 and fig. 20.1(page 650; page 641).

As to claim 38, Weiss teaches the claimed limitation "wherein the sorted tree structure updates from leaf node level to root node level" as fig. 20.9 and fig. 20.10 (page 646).

As to claim 47, Weiss teaches the claimed limitation "wherein the step of redistributing the contiguous empty nodes maintains the invariants of the sorted tree structure" as (pages 651, lines 5-22).

As to claim 48, Weiss teaches the claimed limitation "wherein the step of redistributing the contiguous empty nodes maintains a consistent lookup operation on the sorted tree structure" as (page 648, lines 14-34).

4. Claim 46 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mark ALLen Weiss, Algorithms, Data Structures, and Problem Solving With C++ 1996 in view of Leenstra, Sr. et al (USP 5303367).

As to claim 46, Weiss discloses the claimed limitation subject matter in claim 38, except the claimed limitation " wherein the sorted tree structure is reverse sorted".

However, Weiss teaches that a tree is sorted in (fig. 20. 3, page 642). Also, Leenstra teaches that the linked Data Sets are maintained in inverted hierarchical sorted order relative to the Key Data Set at all times (fig. 10, col. 8, lines 35-42).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Leenstra's teaching of maintaining linked data sets in inverted hierarchical sorted order to Weiss's system in order to allow a user can sort a hierarchy data in any order.

5. Claim 46 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mark ALLen Weiss, Algorithms, Data Structures, and Problem Solving With C++ 1996 in view of Layden et al (or hereinafter "Layden") (USP 5560006).

As to claim 46, Weiss discloses the claimed limitation subject matter in claim 38, except the claimed "wherein the sorted tree structure is reverse sorted".

Layden teaches an inverted binary tree (col. 5, lines 11-12).

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Layden's teaching of inverted binary treeto Weiss's system in order to allow a user can sort a hierarchy data in any order.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cam-Y Truong whose telephone number is (703-605-1169). The examiner can normally be reached on Mon-Fri from 8:00AM to 4:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene, can be reached on (703-305-9790). The fax phone numbers for the organization where this application or proceeding is assigned is (703)-872-9360.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703-305-3900).

Cam-Y Truong

2/19/04

SHAHID ALAM
PRIMARY EXAMINER